Sex: The gender of the person, where 1 represents male and 0 represents female.

Target: A binary variable where 1 represents the presence of a heart disease and 0 represents the absence of a heart disease.

```
import pandas as pd
import numpy as np
df = pd.read_csv("/content/1645792390_cep1_dataset (2).csv")
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
df.shape
```

```
(303, 14)
```

```
df.columns
```

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

```
df.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

```
df.duplicated().sum()
```

```
1
```

```
# Remove a single duplicated row
df.drop_duplicates(keep="first", inplace=True)
```

```
df.duplicated().sum()
```

```
0
```

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope

```
cat = df.select_dtypes(include=["object"]).columns.tolist()
cat
```

```
[]
```

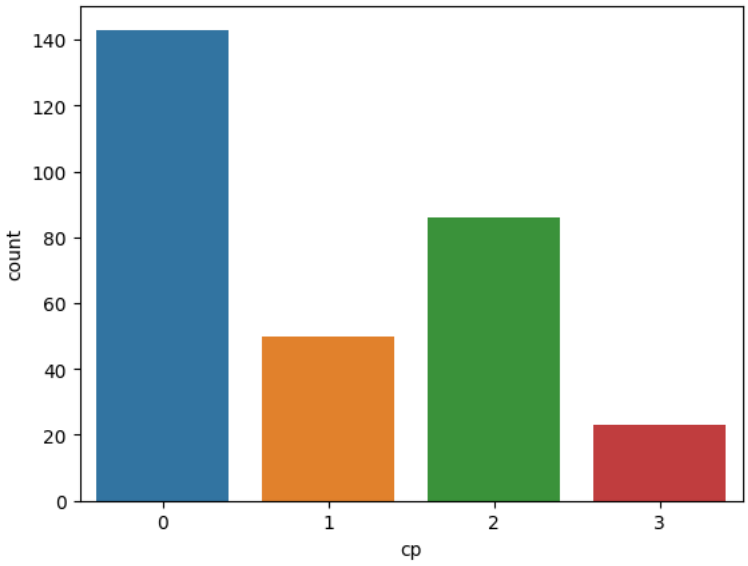
min	29.00000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000

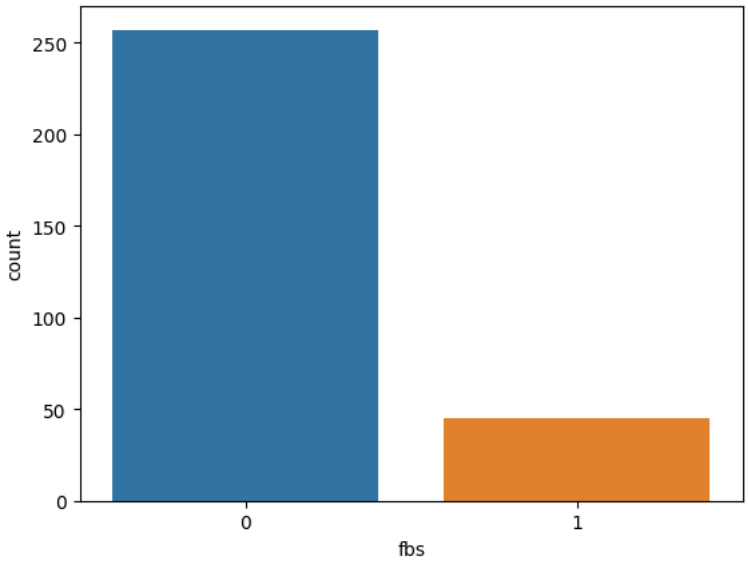
Hence there is no any categorical type data column in the dataset.

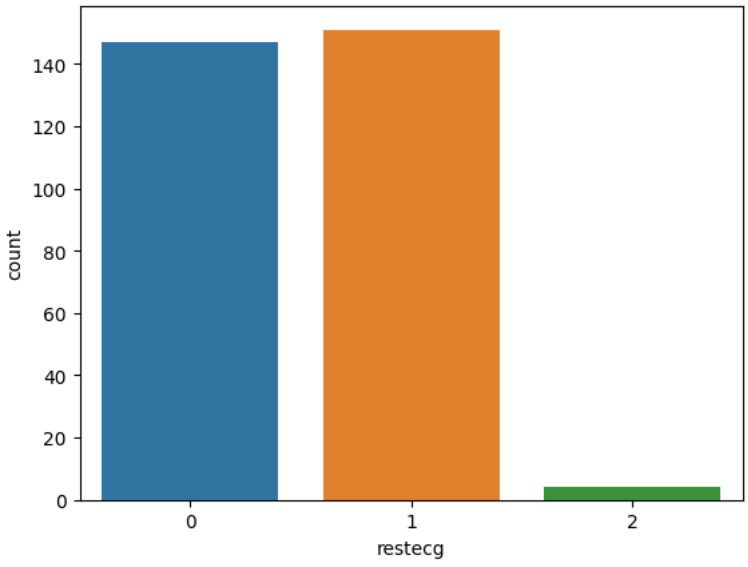
50%	55.50000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	153.500000	0.000000	0.000000	1.000000

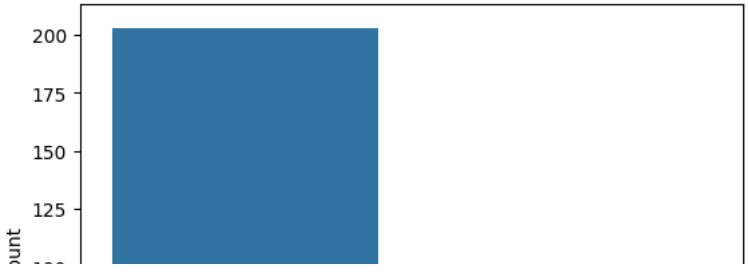
```
import seaborn as sns
import matplotlib.pyplot as plt

columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal']

# Plot count plots for all columns
for col in columns:
    sns.countplot(x=col, data=df)
    plt.show()
```

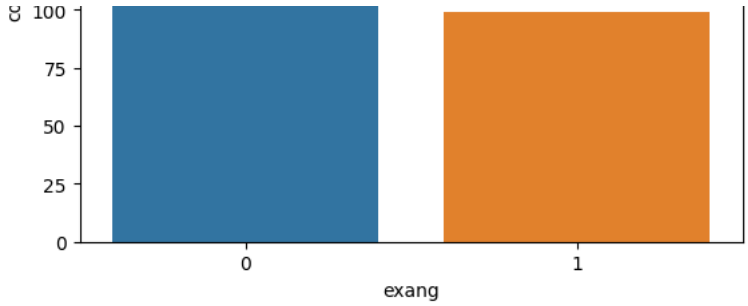

0
0
1
sex

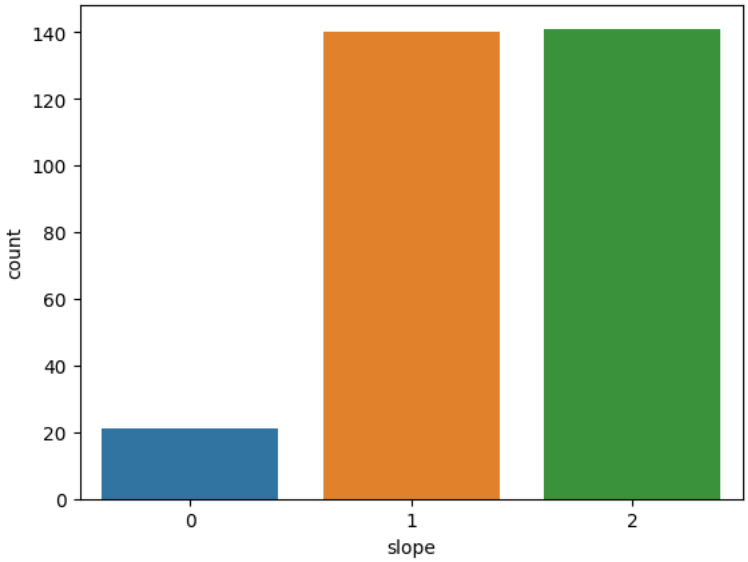

140
120
100
80
60
40
20
0
count
0
1
2
3
cp


250
200
150
100
50
0
count
0
1
fbs


140
120
100
80
60
40
20
0
count
0
1
2
restecg

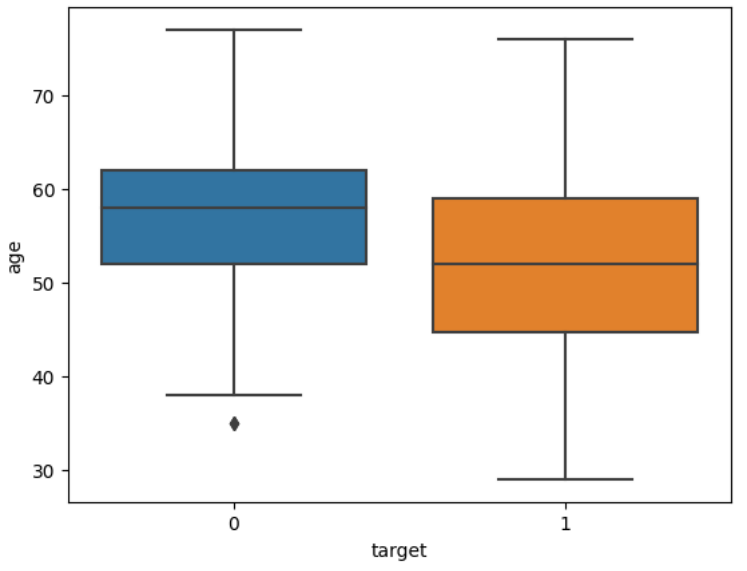

200
175
150
125





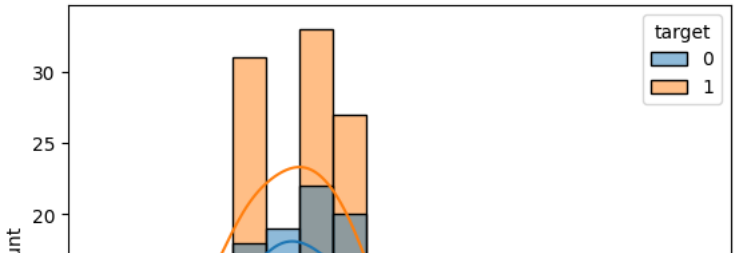

```
sns.boxplot(x="target", y="age", data=df)
```

```
<Axes: xlabel='target', ylabel='age'>
```



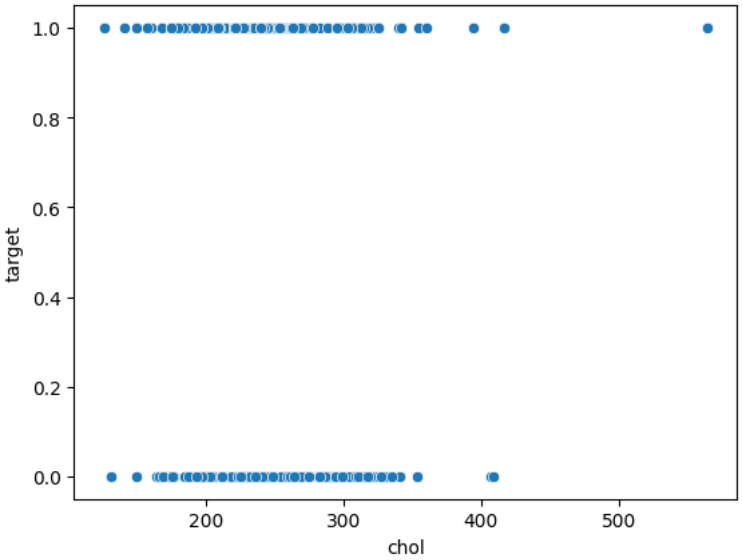

```
sns.histplot(data=df, x="trestbps", hue="target", kde=True)
```

```
<Axes: xlabel='trestbps', ylabel='Count'>
```



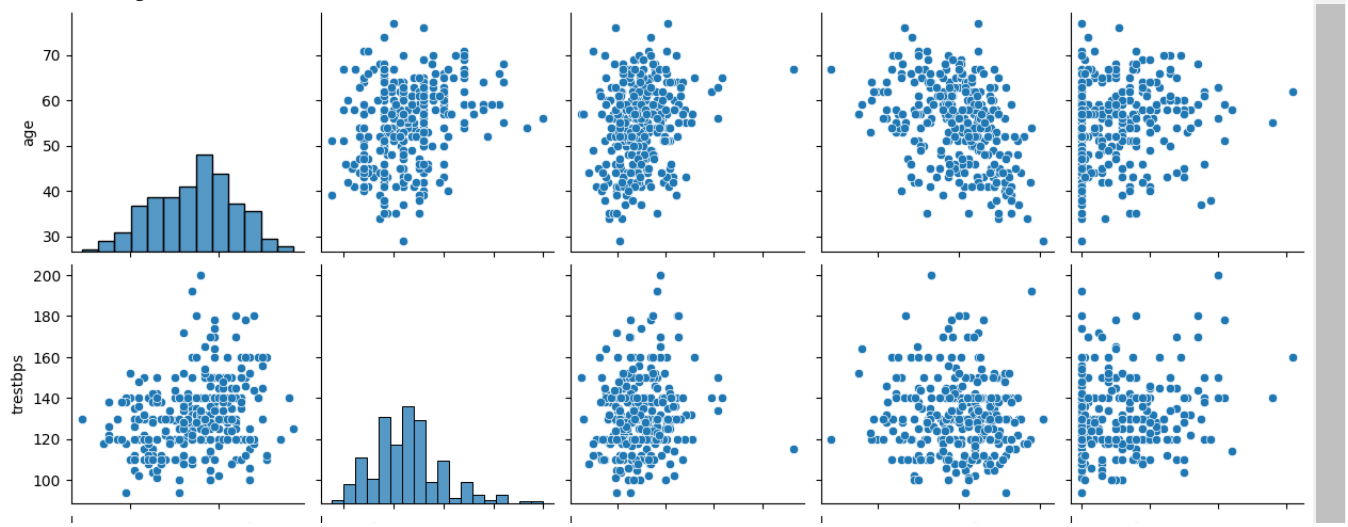

```
sns.scatterplot(x="chol", y="target", data=df)
```

```
<Axes: xlabel='chol', ylabel='target'>
```



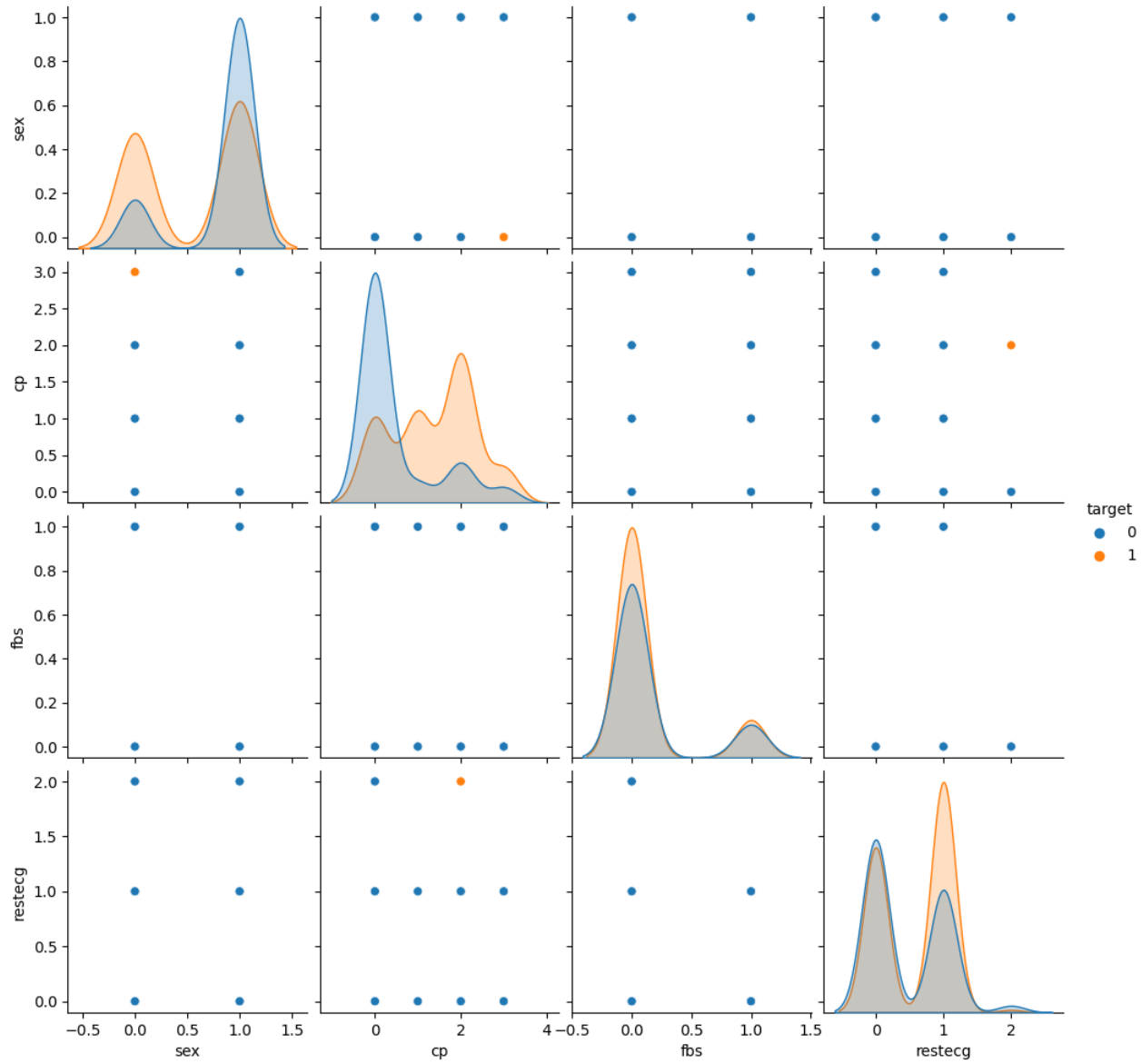

```
sns.pairplot(data=df,·vars=["age",·"trestbps",·"chol",·"thalach",·"oldpeak"])
```

```
<seaborn.axisgrid.PairGrid at 0x7f7f08e93880>
```



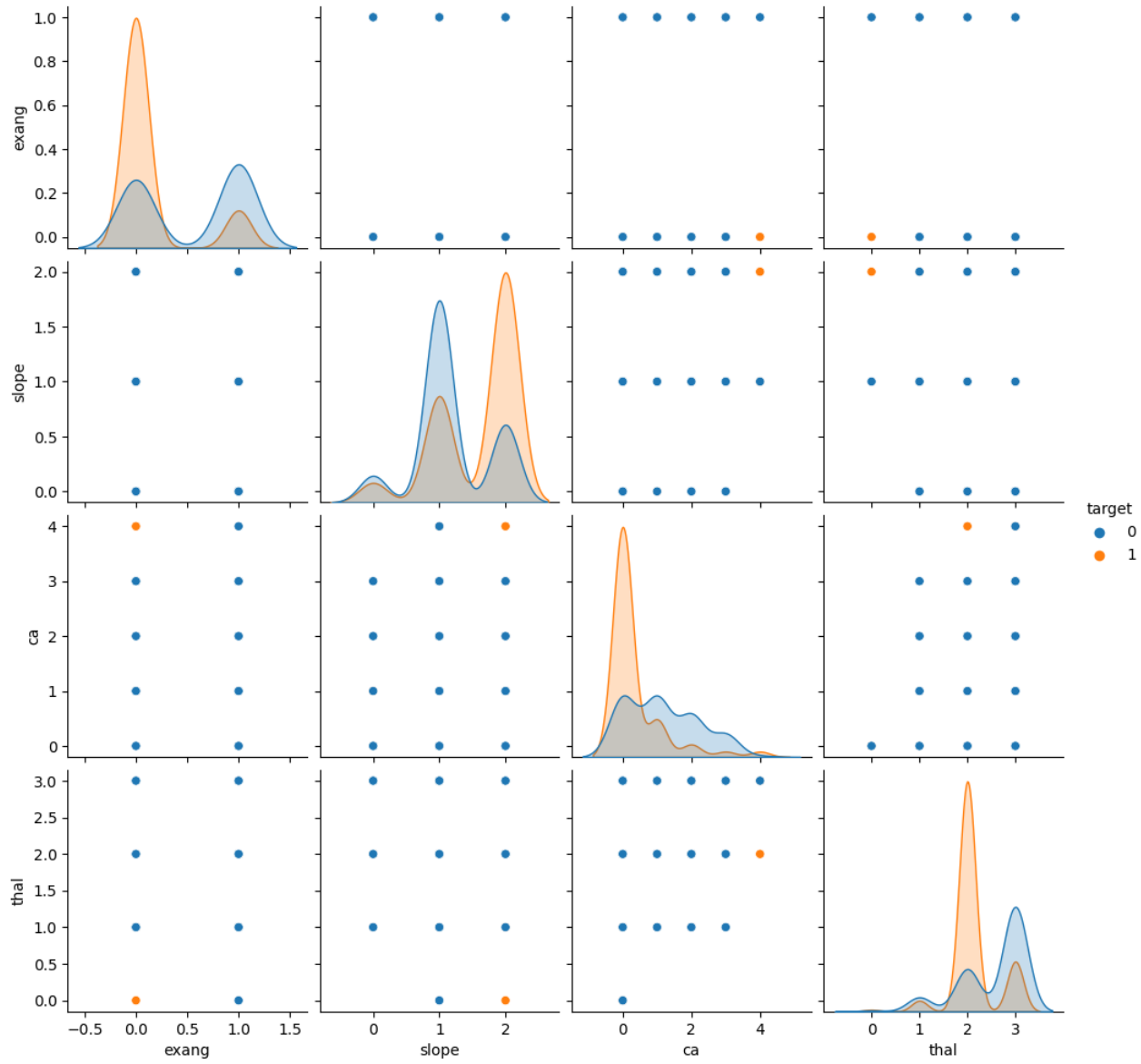

```
sns.pairplot(data=df, vars=["sex", "cp", "fbs", "restecg"], hue="target")
```

```
<seaborn.axisgrid.PairGrid at 0x7f7f0451a730>
```




```
sns.pairplot(data = df, vars = ["exang", "slope", "ca", "thal"], hue = 'target')
```

```
<seaborn.axisgrid.PairGrid at 0x7f7f01ebd2e0>
```




```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Split the data into training and testing sets
X = df.drop("target", axis=1)
y = df["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=10)


# Preprocess the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a logistic regression model
clf = LogisticRegression(random_state=42)
clf.fit(X_train, y_train)

# Evaluate the model on the testing set
y_pred = clf.predict(X_test)
print(y_pred)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[0 1 0 1 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1
 0 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1]
```

```
[[23 10]
 [ 3 25]]
              precision    recall  f1-score   support

           0       0.88      0.70      0.78        33
           1       0.71      0.89      0.79        28

    accuracy                           0.79        61
   macro avg       0.80      0.79      0.79        61
weighted avg       0.81      0.79      0.79        61
```

Colab paid products - Cancel contracts here

0s completed at 10:11 PM